# CREAT

Potential for unintentional file deletion and unstable race conditions

Sean Barnum, Cigital, Inc. [vita[1]]



2007-03-20

## Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 9181 bytes

| | |
|---|---|
| **Attack Category** | • Path spoofing or confusion problem<br>• Malicious Input |
| **Vulnerability Category** | • TOCTOU - Time of Check, Time of Use<br>• Privilege escalation problem<br>• Indeterminate File/Path |
| **Software Context** | • File Creation |
| **Location** | • fcntl.h |
| **Description** | The creat function creates a new ordinary file or prepares to rewrite an existing file named by the path name pointed to by path.<br>If the file exists, the length is truncated to 0 and the mode and owner are unchanged.<br>If the file does not exist, the file's owner ID is set to the effective user ID of the process. The group ID of the file is set to the effective group ID of the process, or if the S_ISGID bit is set in the parent directory then the group ID of the file is inherited from the parent directory. The access permission bits of the file mode are set to the value of mode modified as follows:<br>• If the group ID of the new file does not match the effective group ID or one of the supplementary group IDs, the S_ISGID bit is cleared.<br>• All bits set in the process's file mode creation mask (see umask(2) ) are correspondingly cleared in the file's permission mask.<br>• The "save text image after execution bit" of the mode is cleared (see chmod(2) for the values of mode).<br>Upon successful completion, a write-only file descriptor is returned and the file is open for writing, even if the mode does not permit writing. The file pointer is set to the beginning of the file. The file descriptor is set to remain open across exec functions |

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

(see fcntl(2) ). A new file may be created with a mode that forbids writing.

The call creat(path, mode) is equivalent to

open(path, O_WRONLY | O_CREAT | O_TRUNC, mode)

This function is a problem, because it is possible to unintentionally delete a file or enter a potentially unstable race condition.

creat() is vulnerable to TOCTOU attacks. The existence of a call to this function should be flagged regardless if a "check" function precedes it.

**APIs**

| FunctionName | Comments |
|---|---|
| _creat | use |
| _wcreat | use |
| creat | use |

**Method of Attack**

The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results.

The creat() call is a use-category call, which when preceded by a check-category call can be indicative of a TOCTOU vulnerability.

A TOCTOU attack in regards to creat() can occur when

a. A check for existence of directory occurs

b. The directory is created

Between a and b, an attacker could, for example, link the target directory (the directory to be created) to a known directory. The subsequent creation of the directory would either fail or have unexpected results and or behavior.

**Exception Criteria**

**Solutions**

| Solution Applicability | Solution Description | Solution Efficacy |
|---|---|---|
| Generally applies to most occurrences of creat(). | The most basic advice for TOCTOU vulnerabilities is to not | Does not resolve the underlying vulnerability but limits the |

| | | | |
|---|---|---|---|
| | | perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.<br><br>General usage of creat() should not need a check. | false sense of security given by the check. |
| | Generally applies to most occurrences of creat(). | Limit the interleaving of operations on files from multiple processes. | Does not eliminate the underlying vulnerability but can help make it more difficult to exploit. |
| | Generally applies to most occurrences of creat(). | Limit the spread of time (cycles) between the check and use of a resources | Does not eliminate the underlying vulnerability but can help make it more difficult to exploit. |
| | Generally applies to most occurrences of creat(). | Recheck the resource after the use call to verify that the action was taken appropriately. | Effective in some cases. |
| **Signature Details** | #include "sys/types.h"<br>#include "sys/stat.h"<br>#include "fcntl.h"<br><br>int creat (const char *path, mode_t mode) | | |
| **Examples of Incorrect Code** | `#include "fcntl.h"`<br><br>`int check_status;` | | |

```
int use_status;
struct stat statbuf;

check_status=stat("tobecreateddir",
&statbuf);
[...]

mode_t mode = S_IRUSR | S_IWUSR |
S_IRGRP | S_IROTH;

[...]

fd = creat("/tmp/file", mode);
```

**Examples of Corrected Code**

The following example creates the file /tmp/file with read and write permissions for the file owner and read permission for group and others. The resulting file descriptor is assigned to the fd variable.

```
#include <fcntl.h>
...
int fd;
mode_t mode = S_IRUSR | S_IWUSR |
S_IRGRP | S_IROTH;

...
fd = creat("/tmp/file", mode);
...
```

**Source References**

- Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, pp 220, 222
- UNIX man page for creat()
- Microsoft Developer Network Library (MSDN)
- http://bama.ua.edu/cgi-bin/man-cgi?creat+2
- http://www.cigital.com/papers/download/bsi7-knowledge.pdf

**Recommended Resource**

**Discriminant Set**

| | |
|---|---|
| **Operating Systems** | • UNIX<br>• Windows |
| **Languages** | • C<br>• C++ |

## Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com